

Handling Churn in DHT

S. Rea, D. Geels, T. Roscoe and J. Kubiatowicz

Presented 17.11.2004 by Joakim Koskela

Introduction

Presented through

- Examining real-life P2P behaviour, deriving requirements for DHT:s from these
- Presenting the design of a churn-resistant DHT, Bamboo
- Examining the performance of Bamboo and other existing DHT:s in simulated real-life conditions

Terms

- *Lifetime* – the time between when the node enters and permanently leave the network
- *Session time* – duration of a constant connection to the network
- *Availability* – the session time / lifetime ratio

Applications depend on the routing level which is sensitive to the length of session times

Session time observations

Results vary

- Median between one minute and one hour

First Author	Systems Observed	Session Time
Saroiu [21]	Gnutella, Napster	50% \leq 60 min.
Chu [7]	Gnutella, Napster	31% \leq 10 min.
Sen [22]	FastTrack	50% \leq 1 min.
Bhagwan [3]	Overnet*	50% \leq 60 min.
Gummadi [11]	Kazaa	50% \leq 2.4 min.

Conclusion

- A P2P network build on a DHT should be able to cope with session times as low as one minute

Bamboo - Pastry

- Built on Pastry
- Each node has a 160 bit, *random*, identifier
- Each node maintain a *leaf* set of the $2k$ closest nodes
- Each node has also a multileveled routing table
- Recursive routing
 - Each node communicates only with a specific set of peers
 - Routes primary using the leaf set, falls back on the routing table

Joining the network

Based on Pastry's join operation:

```

(1) function join ( $A, G$ ) =
(2)    $G' =$  nearest_neighbor ( $A, G$ );
(3)    $B, P =$  lookup ( $G', ID_A$ );
(4)    $L =$  get_leaf_set ( $A$ );
(5)   for  $i$  from 0 to  $|P| - 1$ 
(6)      $l =$  length_of_longest_matching_prefix ( $ID_A, ID_{P_i}$ );
(7)      $R_i = P_i.get\_routing\_table\_level (l)$ 

```

Bamboo does not try to generate an optimal routing table

- Acquires only the set of leaf nodes and uses the nodes on route P in creating an initial routing table
- Avoids the network measurement problem

Routing table optimized during use

- Constant RTT / RTO calculation

Optimization processes

- Leaf set update

Nodes send periodically its leaf set to another node

- Nodes reply with their own leaf set

Epidemic distribution of node (routing) information

- Routing table tuning mechanisms

Local tuning

- Node requests from a neighbor all its entries at some level in the routing table
- Best suited in near-optimal situations

Global tuning

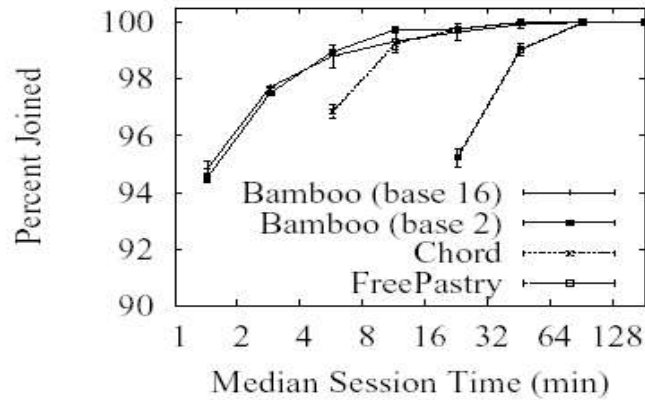
- Message to a random node at some entry in the routing table

Failure detection

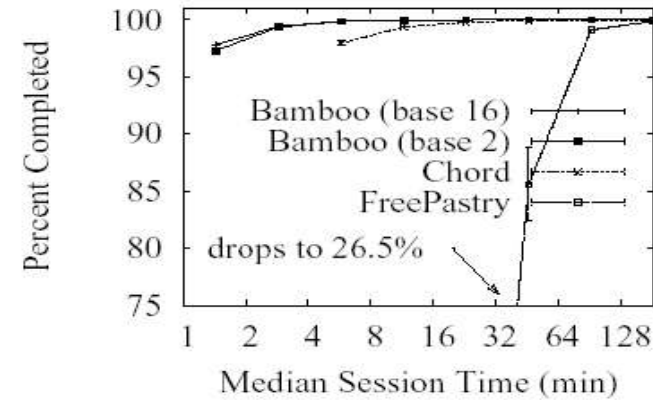
“Churn is the norm”

- Timeout values important
 - Too fast: churn, resulting in congestion
 - Too slow: not enough information about nodes to route
- Bamboo constantly calculates timeout values
- Possible as bamboo uses recursively routing

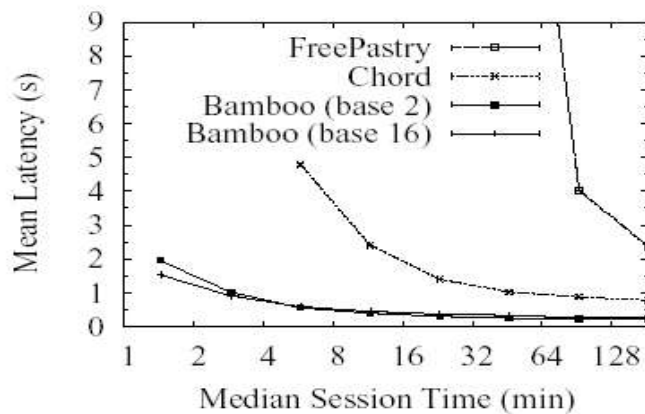
Results – successful joins, lookups



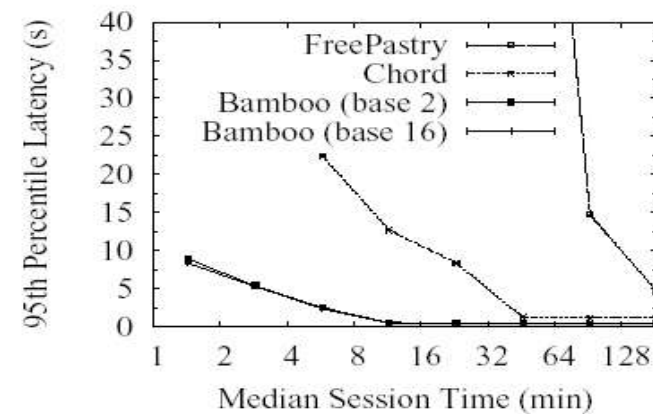
(a)



(b)



(d)

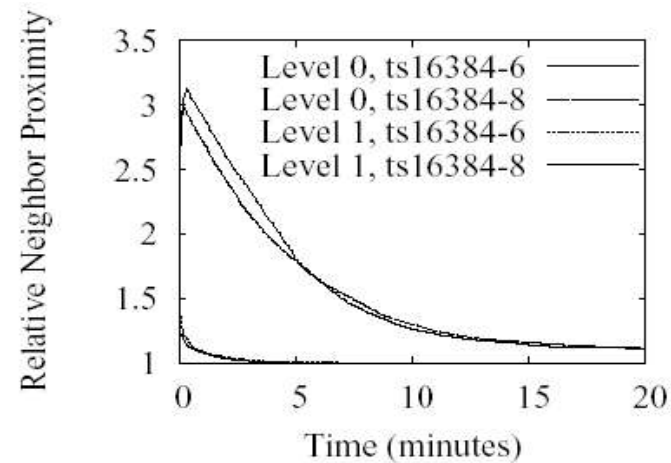
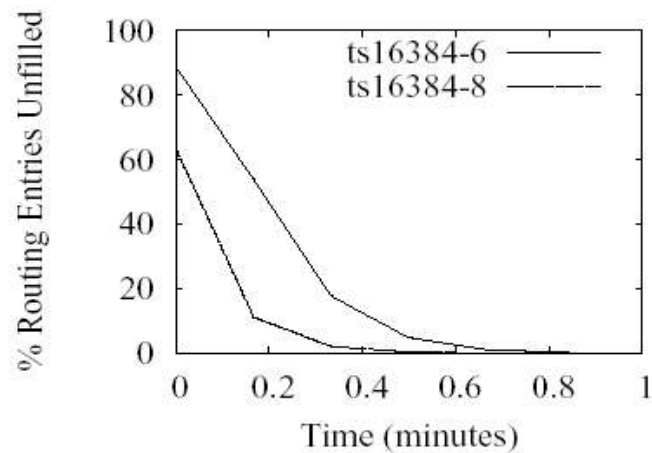
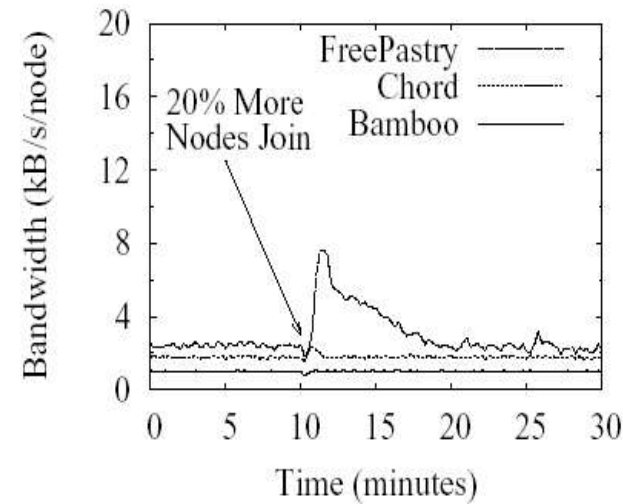
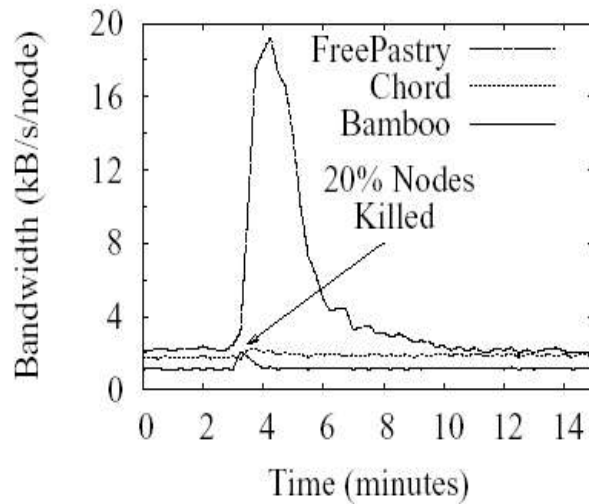


(e)

Bamboo performs clearly better than the other

- Pastry has trouble with session times < 30min

Results – recovery, initiation



Proactive may lead to a churn-loop

Conclusions

Churn has not been considered enough in previous DHTs

- Mean session times seen in deployed P2P networks hard to handle
- Failure detection important

Bamboo

- Designed to survive churn
- Good performance in low-churn networks as well

Shows maturity in DHTs

- One step closer to being used in real-life